

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-073449

(43)Date of publication of application : 26.03.1993

(51)Int.Cl.

G06F 13/00

G06F 12/00

(21)Application number : 03-231332

(71)Applicant : KOBE NIPPON DENKI SOFTWARE KK

(22)Date of filing : 11.09.1991

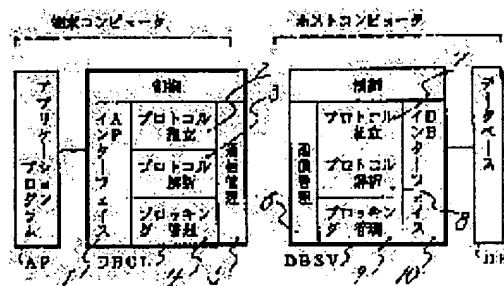
(72)Inventor : KOMATSU MICHIO

(54) DATA PROCESSING SYSTEM FOR DATA BASE CLIENT SERVER SYSTEM

(57)Abstract:

PURPOSE: To operate the whole system at a high speed by decreasing the number of times of communication between a terminal and a host computer, in the system for executing a data processing of a data base connected to a data base server system on a host computer through a data base client system.

CONSTITUTION: In the case a data base client system (DBCL) transfers data operation language to a data base server system (DBSV), the data operation language is subjected to blocking in order to request a series of processings in a lump, and in the case the (DBCL) transfers its data operation language subjected to blocking, when it is that which exceeds a communicable buffer size, it is transferred repeatedly, and in the case the (DBSV) returns a result of data base processing and a record, the result of data base processing and the record are subjected to blocking, and in the case the data base server system transfers those result of data base processing and record subjected to blocking, if they are those which exceed a communicable size, the transfer is continued repeatedly.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

BEST AVAILABLE COPY

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平5-73449

(43)公開日 平成5年(1993)3月26日

(51)Int.Cl. ⁵	識別記号	庁内整理番号	FI	技術表示箇所
G 0 6 F 13/00	3 5 1 E	7368-5B		
12/00	5 4 5 M	8944-5B		

審査請求 未請求 請求項の数1(全 12 頁)

(21)出願番号 特願平3-231332

(22)出願日 平成3年(1991)9月11日

(71)出願人 000192545

神戸日本電気ソフトウェア株式会社
兵庫県神戸市西区高塚台5丁目3番1号

(72)発明者 小松 道代

兵庫県神戸市西区高塚台5丁目3番1号神
戸日本電気ソフトウェア株式会社内

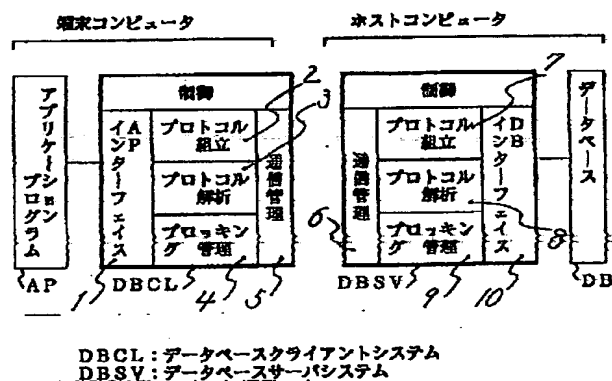
(74)代理人 弁理士 内原 晋

(54)【発明の名称】 データベースクライアント・サーバシステムのデータ処理方式

(57)【要約】

【目的】端末コンピュータ上のアプリケーションプログラムが、データベースクライアントシステムを介して、ホストコンピュータ上のデータベースサーバシステムと連結されたデータベースのデータ処理を行うシステムで、端末・ホストコンピュータ間の通信回数を減らすことにより、システム全体を高速化しようとするものである。

【構成】データベースクライアントシステムがデータ操作言語をデータベースサーバシステムに転送する場合、一連の処理を一括して要求するためにデータ操作言語をブロッキングし、データベースクライアントシステムがそのブロッキングしたデータ操作言語を転送する場合、通信可能なバッファサイズを越えるのであれば、繰り返し転送し、データベースサーバシステムがデータベース処理結果やレコードを返却する場合、データベース処理結果やレコードをブロッキングし、データベースサーバシステムがそのブロッキングしたデータベース処理結果やレコードを転送する場合、通信可能なバッファサイズを越えるのであれば、繰り返し転送し続ける。



【特許請求の範囲】

【請求項1】 端末コンピュータ上のアプリケーションプログラムが、データベースクライアントシステムを介して、ポストコンピュータ上のデータベースサーバシステムと連結されたデータベースのデータ処理を行う方式において、

前記データベースクライアントシステムがデータ操作言語を前記データベースサーバシステムに転送する場合に一連の処理を一括して要求するためにデータ操作言語をブロッキングして転送し、ブロッキングしたデータ操作言語が通信可能なバッファサイズを越えるものであれば、通信可能なバッファサイズにブロッキングしたデータ操作言語を繰り返し転送し続ける手段と、前記データベースサーバシステムがデータベース処理結果またはレコードを返却する場合、データベース処理結果およびレコードをブロッキングして転送し、ブロッキングしたデータベース処理結果およびレコードが通信可能なバッファサイズを越えるものであれば、通信可能なバッファサイズにブロッキングしたデータベース処理結果およびレコードを繰り返し転送し続ける手段と、前記データベースクライアントシステムがデータベースサーバシステムから受信したデータベース処理結果およびレコードをデブロッキングする手段と、前記データベースサーバシステムが前記データベースから受信したデータ操作言語をデブロッキングする手段とを含むことを特徴とするデータベースクライアント・サーバシステムのデータ処理方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、データ処理を行う端末コンピュータのアプリケーションプログラム（以後APと略す）から、データベースクライアントシステム（DBCL）を介し、データ操作言語（DML）を転送することにより、ホストコンピュータ上のデータベースサーバシステム（DBSV）に結合したデータベース（DB）を操作する様なシステムのデータ処理方式に関する。

【0002】

【従来の技術】従来のDBCL・DBSV間のデータ処理方式は、DBCLがAPからの要求を受けると即時にDBSVへ転送し、DBSVもDBからデータ処理結果（レスポンス）やレコードを受け取ると即時にDBCLへ転送するという繰り返しであった。

【0003】図3および図5は従来のデータベースクライアント・サーバシステムのデータ処理方式を示す模式図である。

【0004】図3はAPからDBの4件のレコードの読み込み（READ）の処理を示す。DBCLはAPから要求のあったOPEN-DMLをDBSVに転送し、DBSVはDBからのレスポンスをDBCLに送信する。

次に4件のレコードそれぞれについてDBCLはAPから要求されるREADをDBSVに転送し、DBSVはDBのレコードをDBCLに送信する。

【0005】図5はAPからDBへの4件のレコードの書き込み（WRITE）の処理を示す。4件のレコードそれぞれについてDBCLはAPから要求されるWRITEおよびレコードをDBSVに転送し、DBSVはレスポンスをDBCLに送信する。

【0006】

【発明が解決しようとする課題】前述の様に、APは1つの要求に対し1つのレスポンスやレコードしか受け取れないため、頻繁に要求を行う必要があり、そのためDBCL・DBSV間の通信回数が増えていた。

【0007】また、日常のAPでは、レコードの順次読み込み（READ）やレコードの順次書き込み（WRITE）は、特に利用頻度が高く、連続して要求されることが多いDMLである。

【0008】従って、通信回数を減らし、READ・WRITE-DMLのデータ処理時間を短縮することが必須であった。

【0009】

【課題を解決するための手段】本発明のデータベースクライアント・サーバシステムの高速度データ処理方式は、端末コンピュータ上のアプリケーションプログラムが、データベースクライアントシステムを介して、ポストコンピュータ上のデータベースサーバシステムと連結されたデータベースのデータ処理を行う方式において、前記データベースクライアントシステムがデータ操作言語を前記データベースサーバシステムに転送する場合に一連の処理を一括して要求するためにデータ操作言語をブロッキングして転送し、ブロッキングしたデータ操作言語が通信可能なバッファサイズを越えるものであれば、通信可能なバッファサイズにブロッキングしたデータ操作言語を繰り返し転送し続ける手段と、前記データベースサーバシステムがデータベース処理結果またはレコードを返却する場合、データベース処理結果およびレコードをブロッキングして転送し、ブロッキングしたデータベース処理結果およびレコードが通信可能なバッファサイズを越えるものであれば、通信可能なバッファサイズにブロッキングしたデータベース処理結果およびレコードを繰り返し転送し続ける手段と、前記データベースクライアントシステムがデータベースサーバシステムから受信したデータベース処理結果およびレコードをデブロッキングする手段と、前記データベースサーバシステムが前記データベースから受信したデータ操作言語をデブロッキングする手段とを有する。

【0010】

【実施例】本発明について図面を用いて説明する。

【0011】図1は本発明の一実施例のDBCLとDBSVの構成図である。端末コンピュータ上のDBCL

は、APの処理要求を受け付け、またAPにレスポンスやレコードを返却するAPインターフェイス部1、APの処理要求を通信可能な形式に変換するプロトコル組立部2、DBSVから転送されたプロトコルをAP側に渡す形式に変換するプロトコル解析部3、DML、レスポンス、レコードのブロッキング・デブロッキングを行うブロッキング管理部4、そして実際にDBSVとプロトコルを送受信する通信管理部5を有する。

【0012】ホストコンピュータ上のDBSVは、DBCLとプロトコルを送受信する通信管理部6、DBアクセス後のレスポンスやレコードを通信可能な形式に変換するプロトコル組立部7、DBCLから転送されたプロトコルを解析しAPのデータ処理要求を認識するプロトコル解析部8、DML、レスポンス、レコードのブロッキング・デブロッキングを行うブロッキング管理部9、そして実際に、DBアクセス要求を行いレスポンスやレコードを受け取るDBインターフェイス部10がある。

【0013】図2はDB内のファイルをOPENし指定レコード件数(ここでは4件)READする処理の説明図である。

【0014】まず、DBCLはAPから要求を受けた(APインターフェイス1)OPEN-DMLをブロッキングしREAD-DMLもブロッキングする(ブロッキング管理部4)。そして、プロトコルを組立て(プロトコル組立部2)、DBSVに転送する(通信管理部5)。このとき送信権(図では★で示す)も渡す。DBSVはプロトコルを受け取り(通信管理部6)、解析する(プロトコル解析部8)。プロトコルはDMLをブロッキングしているため、デブロッキングを行い(ブロッキング管理部9)APの要求を把握する。そして実際にDBにアクセスを行う(DBインターフェイス部10)。

【0015】この例ではまずOPENを行い、OPENのレスポンスをブロッキングする(ブロッキング管理部9)。READを行い(DBインターフェイス部10)レコード1をブロッキングし(ブロッキング管理部9)再びREADを行い(DBインターフェイス部10)レコード2をブロッキング(ブロッキング管理部9)する。そして、ここで通信可能なバッファサイズに達したためプロトコルを組立て(プロトコル組立部7)DBCLへ送信する(通信管理部6)。そして、まだAP指定処理が終了していないため、送信権は渡さないで引き続きREADを行う(DBインターフェイス部10)。レコード3、4をREADした時点でAPの指定処理が終了するため、プロトコルを組立て(プロトコル組立部7)DBCLへ送信する(通信管理部6)。そしてこの時、送信権はDBCLへ渡す。

【0016】図3は図2と同じ処理を従来の方で行った例である。APで受け取ったレコードは同じ件数でも、通信回数は図2で3回、従来の場合の図3は10回

で本実施例の場合の3倍である。

【0017】図4は本実施例でWRITE-DMLのレコードをブロッキングして数回に渡って送信する例である。

【0018】まず、DBCLはAPから要求を受けた(APインターフェイス部1)WRITE-DMLをブロッキングしレコードもブロッキングする(ブロッキング管理部4)。そして、送信バッファが一杯になるとプロトコルを組立て(プロトコル組立部2)、DBSVに転送する(通信管理部5)。この時、次に続くWRITE-DMLがあるため送信権は渡さない。2回WRITE-DMLとレコードもブロッキングしAPからの要求が終わったところでプロトコルを組立て(プロトコル組立部2)、DBSVに転送する(通信管理部5)。この時は送信権も渡す。DBSVはプロトコルを受け取り(通信管理部6)、解析する(プロトコル解析部8)。プロトコルはDMLをブロッキングしているため、デブロッキングを行い(ブロッキング管理部9)APの要求を把握する。そして実際にDBにアクセスを行う(DBインターフェイス部10)。この例ではまずWRITEを行い、レスポンスをブロッキングする(ブロッキング管理部9)。続けて、WRITEを行い、レスポンスのブロッキングをし(ブロッキング管理部9)、プロトコルを組立て(プロトコル組立部7)DBCLへ送信する(通信管理部6)。この時、送信権も渡す。

【0019】図5は図4と同じ処理を従来の方で行った例である。WRITEしたレコードは同じ件数でも、通信回数は図4で3回、従来の場合の図5は8回で本実施例の場合の2倍以上である。

【0020】これらの制御は主にブロッキング管理部4、9で行う。ブロッキング管理部4ではDMLのブロッキング、レスポンスやレコードのデブロッキング、ブロッキング管理部9ではDMLのデブロッキング、レスポンスやレコードのブロッキングを行う。

【0021】図6、図7のフローチャートではDBCL・DBSVそれぞれの処理の流れを示す。ステップ601~613がDBCLの処理を示し、ステップ701~711がDBSVの処理を示す。

【0022】DBCLは、APから要求を入力し(ステップ601)ブロッキングする(ステップ602)。AP要求を終わり又は通信バッファが一杯になると(ステップ603)、プロトコルを組み立てる(ステップ604)。AP要求が終わりでなければ(ステップ605)プロトコルのみを送信し(ステップ606)、AP要求の入力(ステップ601)に戻る。終わりであれば送信権も送信し(ステップ607)DBSVからのプロトコルの受信状態(ステップ608)に入る。

【0023】DBSVはDBCLから送られたプロトコルを受信する(ステップ701)。送信権を受けとると(ステップ702)プロトコルを分解する(ステップ7

03)。AP要求を認識を行い(ステップ704)DBをアクセスし(ステップ705)その結果をブロッキングする(ステップ706)。AP要求の終わり又は通信バッファ一杯になると(ステップ707)プロトコルを組み立て(ステップ708)、AP要求の終わりでなければ(ステップ709)プロトコルのみを送信し(ステップ710)、AP要求認識(ステップ704)に戻り繰り返す。終わりであれば送信権も送信し(ステップ711)DBCLからの受信状態(ステップ701)に戻る。

【0024】DBCLはDBSVから送られたプロトコルを受信し(ステップ608)送信権を受け取ると(ステップ609)プロトコルを分解する(ステップ610)。レスポンスやレコードをデブロッキングして(ステップ611)APへ出力する(ステップ612)。そして結果出力が終わるとAP要求の入力へ戻る(ステップ601)。

【0025】図8は本実施例と従来のデータベースクライアント・サーバシステムのデータ処理方式での処理時間を比較して示すためのタイムチャートであり、左側の2本の棒状の表示は従来の方式での時間を示し、右側の2本のものは本実施例での時間を示す。左右それぞれで左側のものは端末コンピュータ側の時間を示し、右側のものはホストコンピュータ側の時間を示す。図中Rで示す部分はDBCLまたはDBSV内のプロトコル組立等の処理の時間を示し、従来の方式および本実施例の右側の件数は送信したレコードの累積係数を示す。

【0026】図8に示すように本実施例は従来の方式に比べ通信回数を削減したことの他に、通信している間に処理を行っていることも効果のひとつである。

【0027】

【発明の効果】以上説明した様に、本発明によればDBL、レスポンス、レコードのブロッキングによって通信可能バッファサイズを有効利用し、通信回数を大幅に削減し、通信機能の負荷を軽減することで、DBCL・DBSV双方の処理能力を最大限に利用可能とする効果がある。

【図面の簡単な説明】

【図1】本発明のデータベースクライアント・サーバシステムのデータ処理方式を示すブロック図である。

【図2】図1に示す実施例でDBのファイルをOPENしレコードを4件READするデータ処理方式を説明する図である。

【図3】従来のデータベースクライアント・サーバシステムでDBのファイルをOPENしレコードを4件READする処理方式を説明する図である。

10 【図4】図1に示す実施例で4件のレコードをDBへWRITEする処理方式を説明する図である。

【図5】従来のデータベースクライアント・サーバシステムで4件のレコードをDBへWRITEする処理方式を説明する図である。

【図6】図1に示す実施例のフローチャートの一部分である。

【図7】図1に示す実施例の概略フローチャートの一部分である。

20 【図8】図1に示す実施例と従来のデータベースクライアント・サーバシステムのデータ処理方式を比べて示すタイムチャートである。

【符号の説明】

AP アプリケーションプログラム

DB データベース

DBCL データベースクライアントシステム

DBSV データベースサーバシステム

1 APインターフェイス部

2 プロトコル組立部

3 プロトコル解析部

30 4 ブロッキング管理部

5 通信管理部

6 通信管理部

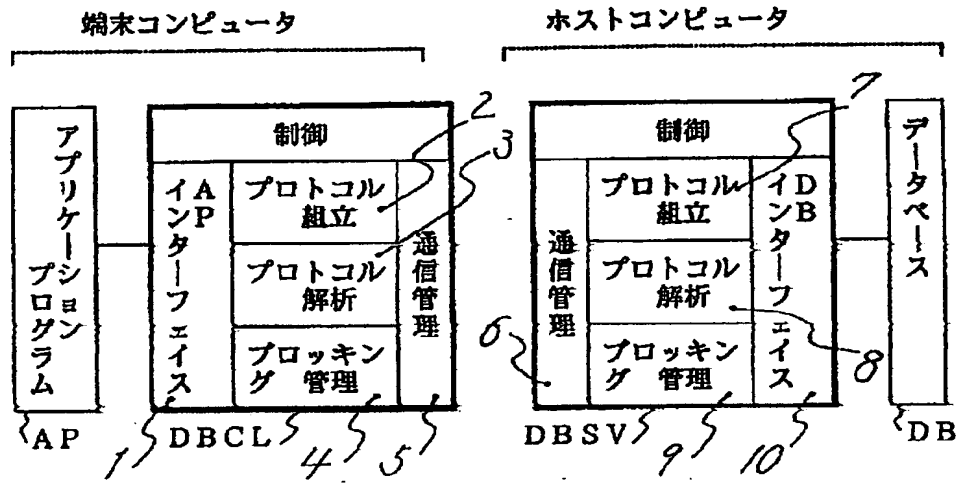
7 プロトコル組立部

8 プロトコル解析部

9 ブロッキング管理部

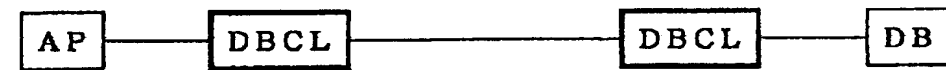
10 DBインターフェイス部

【図1】



DBCL: データベースクライアントシステム
 DBSV: データベースサーバシステム

【図2】



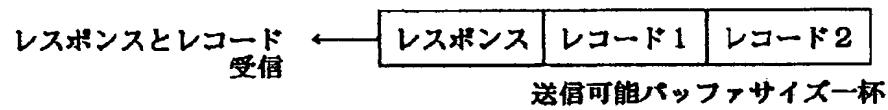
OPEN → ブロッキング
 READ →
 READ →
 READ →
 READ →

ブロッキングしたDML



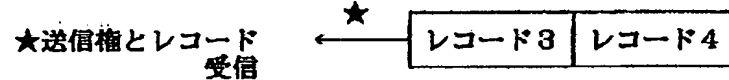
OPEN実行 →
 ← レスポンス
 READ実行 →
 ← レコード1
 READ実行 →
 ← レコード2

ブロッキングしたレスポンスとレコード



READ実行 →
 ← レコード3
 READ実行 →
 ← レコード4

ブロッキングしたレコード



【図3】

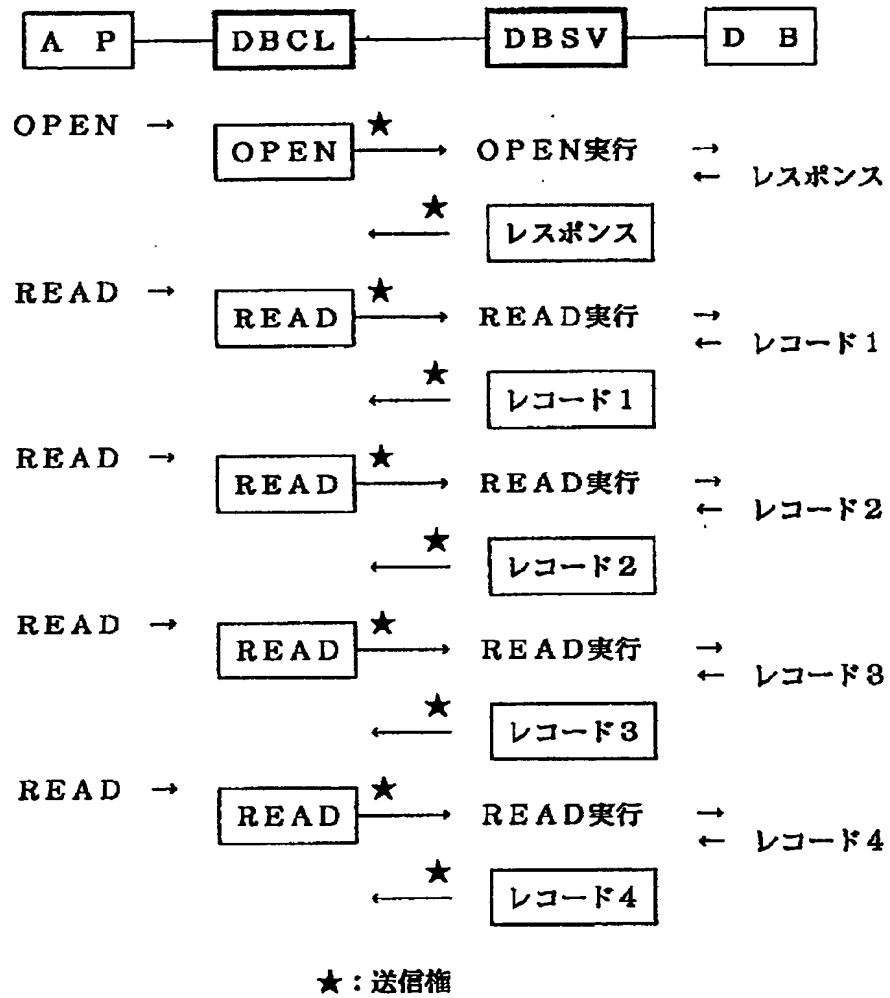


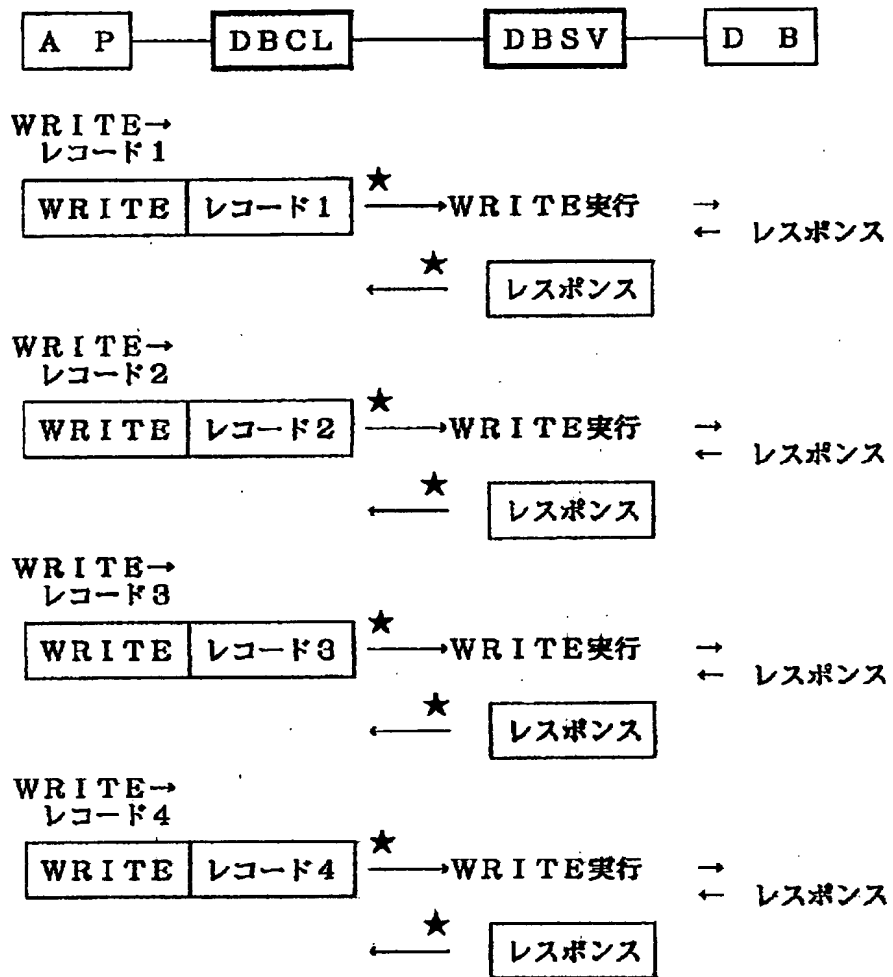
Figure 1 is a sequence diagram illustrating the process of a database access. The diagram shows a sequence of operations: AP, DBCL, DBCL, DB. Below this, it details the execution of OPEN, WRITE, and RECORD operations. OPEN is followed by a blocking period (ブロッキング). WRITE operations are followed by RECORD operations. The sequence is: OPEN, WRITE, RECORD 1, RECORD 2, WRITE, RECORD 3, WRITE, RECORD 4. A star (*) marks the end of the sequence, followed by a response (レスポンス).

```

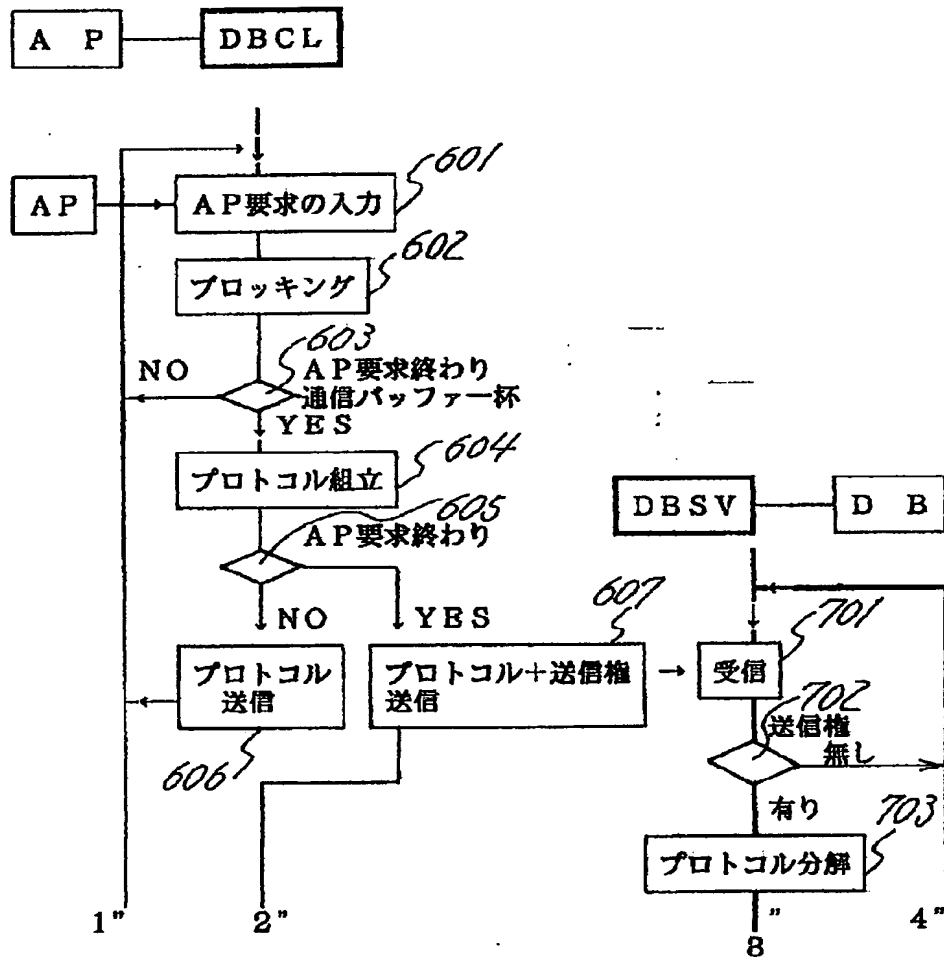
sequenceDiagram
    participant AP
    participant DBCL1 as DBCL
    participant DBCL2 as DBCL
    participant DB
    AP->>DBCL1
    DBCL1->>DBCL2
    DBCL2->>DB
    Note over AP: OPEN → ブロッキング
    Note over AP: WRITE → レコード1
    Note over AP: WRITE → レコード2
    Note over AP: OPEN
    Note over AP: WRITE
    Note over AP: レコード1
    Note over AP: レコード2
    Note over AP: WRITE
    Note over AP: レコード3
    Note over AP: WRITE
    Note over AP: レコード4
    Note over AP: *
    Note over AP: レスポンス
  
```

★：送信櫃

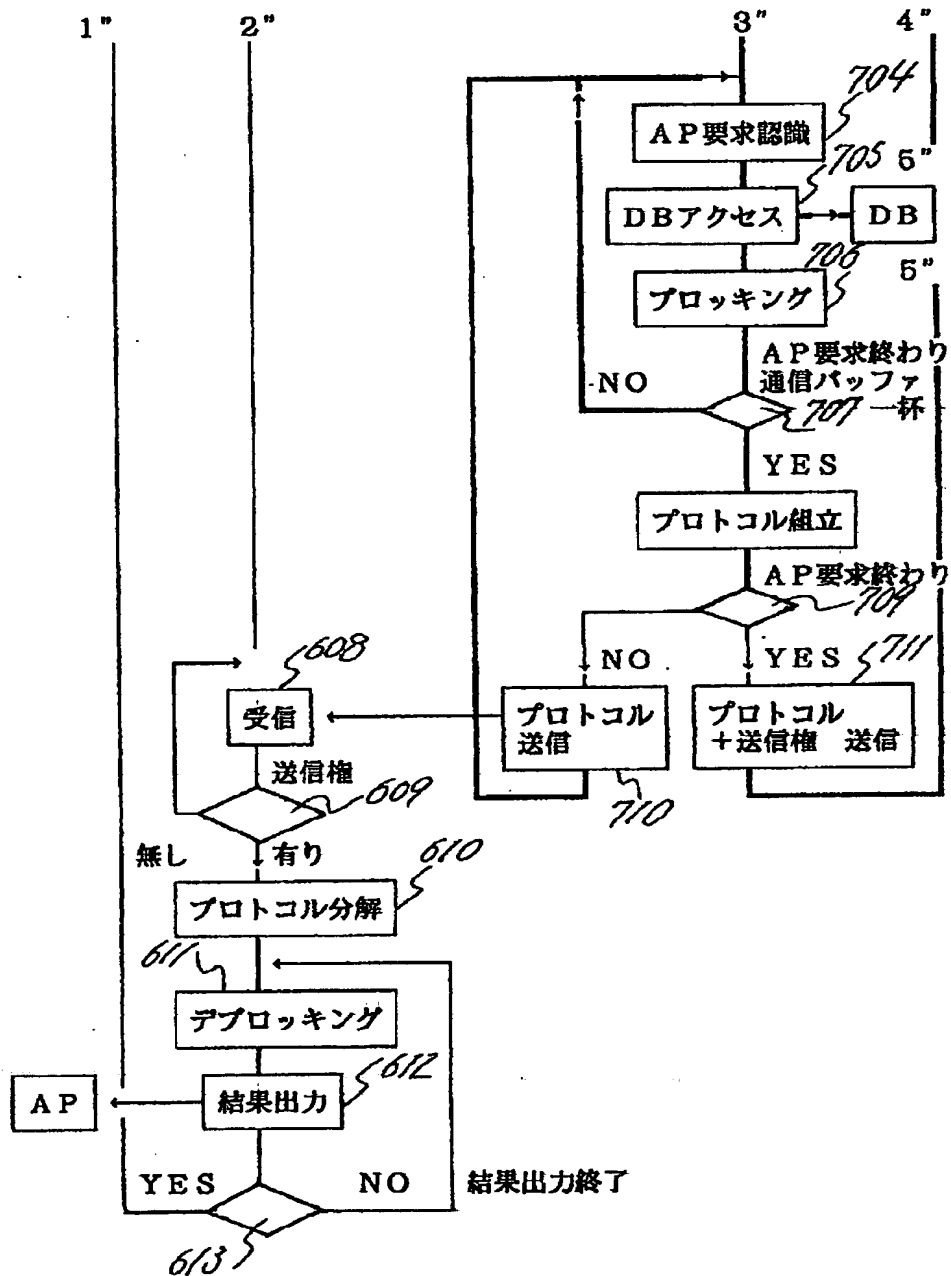
【図5】



【図6】



【図7】



【図8】

従来の方法

本発明

